master ATiAM

Acoustique
Traitement du signal
Informatique
Appliqués à la
Musique

A survey of Machine Learning techniques

Arshia Cont
Musical Representations Team, IRCAM.
cont@ircam.fr

ATIAM 2011-12

http://repmus.ircam.fr/atiam_ml_2011

Artificial Intelligence (AI)

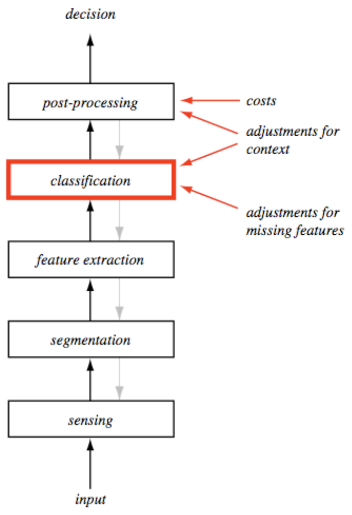
- *What is Artificial Intelligence?*
  by John McCarthy.
  http://www-formal.stanford.edu/jmc/whatisai/
  - "After WWII, a number of people independently started to work on intelligent machines. The English mathematician Alan Turing may have been the first. He gave a lecture on it in 1947. He also may have been the first to decide that AI was best researched by programming computers rather than by building machines. By the late 1950s, there were many researchers on AI, and most of them were basing their work on programming computers."
- Towards *complexity* of real-world structures
  - Ant-colony example
    "The complex behavior of the ant colony is due to the complexity of its environment and not to the complexity of the ants themselves. This suggests the adaptive behavior of learning and representation and the path the science of the artificial should take."
    (H.A. Simons, *The Science of the Artificial,* MIT Press, 1969)

Machine Learning and AI

- Machine Learning deals with sub-problems in engineering and sciences rather than the global "intelligence" issue!
  - Applied
  - A set of well-defined approaches each within its limits that can be applied to a problem set
  - Classification / Pattern Recognition / Sequential Reasoning / Induction / Parameter Estimation etc.
- Our goal today is to *introduce* some well-known and well-established approaches in AI and Machine Learning
  - The methods presented today are not *domain-specific* but for every problem, you start with a design, collect *related data* and then define the learning problem. We will not get into *design* today... .
- Keep in mind that,
  - AI is an *empirical science!*
    - *See "Science of the Artificial" by* H.A. Simons, MIT Press, 1969
  - DO NOT apply algorithms blindly to your data/problem set!
    - The MATLAB Toolbox syndrome: Examine the hypothesis and limitation of each approach before hitting enter!
  - Do not forget your own intelligence!

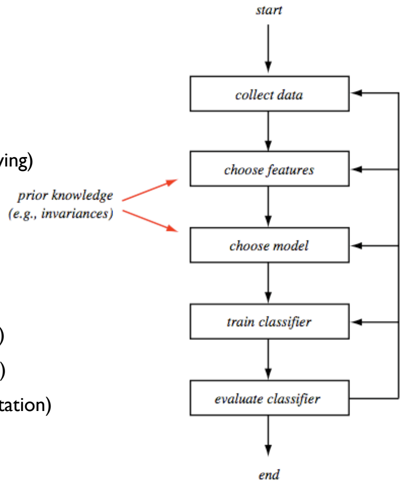
Pattern Recognition

- Pattern recognition in action:
  - Examples:
    - Instrument Classification
    - Audio to Score Alignment (score following)
    - Music genre classification
    - Automatic Improvisation
    - Gesture Recognition
    - Music Structure Discovery
    - Concatenative Synthesis (unit selection)
    - Artist Recovery



Pattern Recognition

- Pattern recognition design cycle:
  - Examples:
    - Instrument Classification
    - Audio to Score Alignment (score following)
    - Music genre classification
    - Automatic Improvisation
    - Gesture Recognition
    - Music Structure Discovery
    - Concatenative Synthesis (unit selection)
    - Room Acoustics (parameter adaptation)
    - Physical Modeling (model learning/adaptation)
    - etc!



start

collect data

choose features

choose model

train classifier

evaluate classifier

end

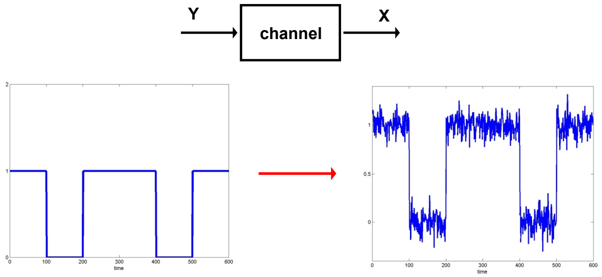
prior knowledge (e.g., invariances)

Machine Learning

- Provide tools and reasoning for the design process of a given problem
- Is an empirical science
- Has a profound theoretical background
- Is extremely diverse
- Should keep you **honest** (and not the contrary!)
- Course objective:
  - To get familiar with Machine Learning tools and reasoning and prepare you for attacking real-world problems in Music Processing

Sample Example (I)

- Communication theory:
  - Question: What should an *optimal decoder* do to recover *Y* from *X* ?
  - *X* is usually referred to as *observation* and is a *random variable.*
  - In most problems, the *real* state of the world (*y*) is not *observable* to us! So we try to *infer* this from the *observation.*



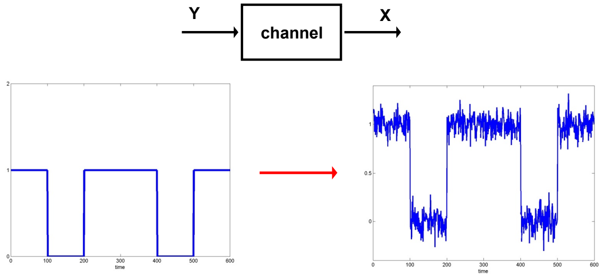
Y

channel

X

Sample Example (I)

- This is a typical ***Classification*** problem
- Intuitive Solution:
  - *Threshold* on 0.5
  - But let's make life more difficult!



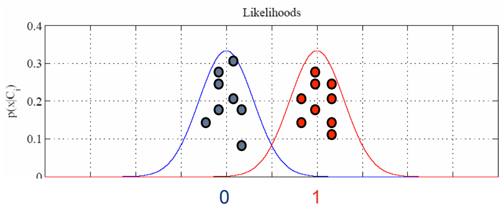
Y

channel

X

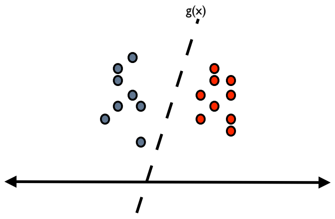
Sample Example (I)

- Simple Solution 1:
  - Define a *decision function* *g(x)* that *predicts* the state of the world (y).
  - and learn it!
- I am thus assuming that the family of *g(x)* that *generate* X if I have Y (the inverse problem).



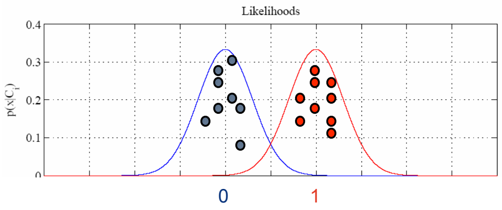
Sample Example (I)

- Simple Solution 2:
  - Try to find an *optimal boundary* (defined as g(x)) that can best separate the two.
  - Define the decision function as + or - distance from this boundry.
- I am thus assuming that the family of *g(x)* that *discriminate* X classes.

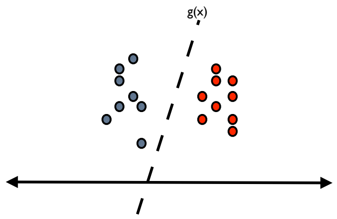


Sample Example (I)

- We just saw two different philosophies to solve our simple problem:
  - Generative Design:

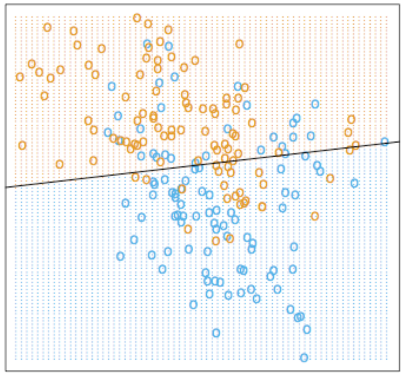


  - Discriminative Design:



Sample Example (I)

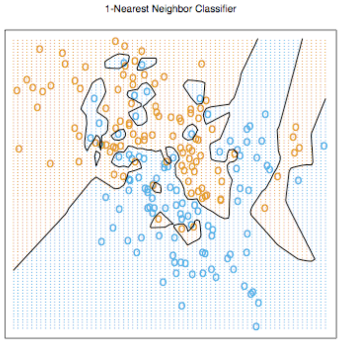
- In the real world things are not as simple
  - Consider the following 2-dimensional problem
  - Not hard to see the problem!



Sample Example (I)

- In the real world things are not as simple
  - Consider the following 2-dimensional problem

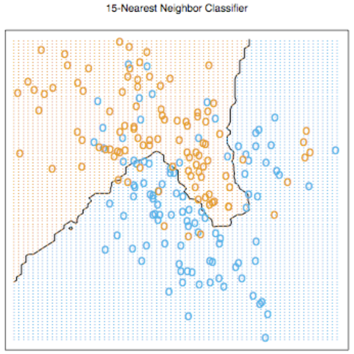
1. To what extend does our solution *generalize* to new data?
   - The central aim of designing a classifier is to correctly classify **novel** input!


1-Nearest Neighbor Classifier

Sample Example (I)

- In the real world things are not as simple
  - Consider the following 2-dimensional problem

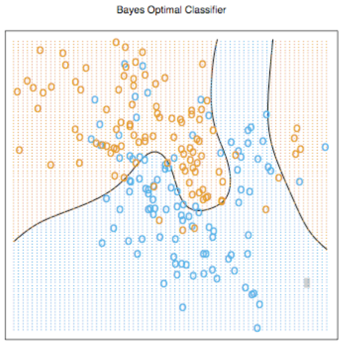
1. To what extend does our solution *generalize* to new data?
   - The central aim of designing a classifier is to correctly classify **novel** input!
2. How do we know when we have collected adequately large and representative set of examples for training?


15-Nearest Neighbor Classifier

Sample Example (I)

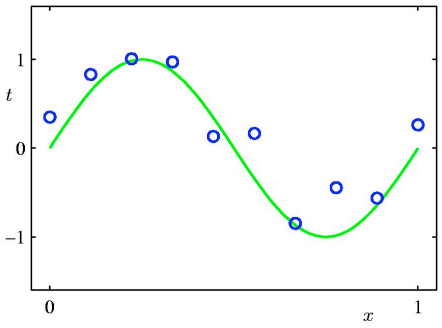
- In the real world things are not as simple
  - Consider the following 2-dimensional problem

1. To what extend does our solution *generalize* to new data?
   - The central aim of designing a classifier is to correctly classify **novel** input!
2. How do we know when we have collected adequately large and representative set of examples for training?
3. How can we decide model complexity versus performance?


Bayes Optimal Classifier

Sample Example (II)

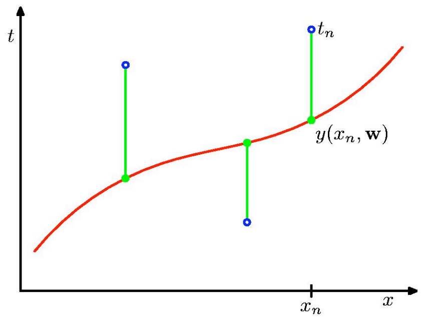
- This is a typical *Regression* problem
- Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sample Example (II)

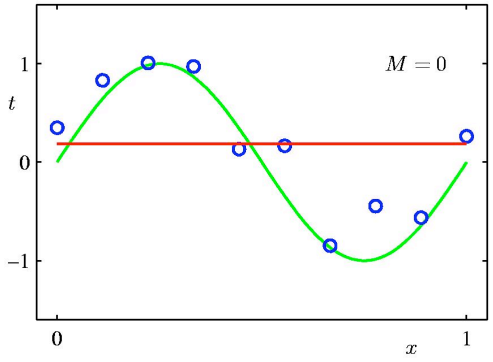
- Polynomial Curve Fitting
  - Sum-of-squares Error Function



$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

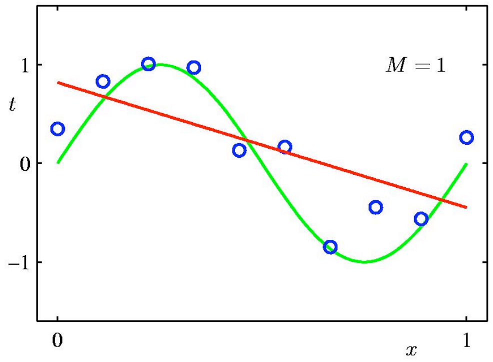
Sample Example (II)

- Polynomial Curve Fitting
  - 0th order polynomial



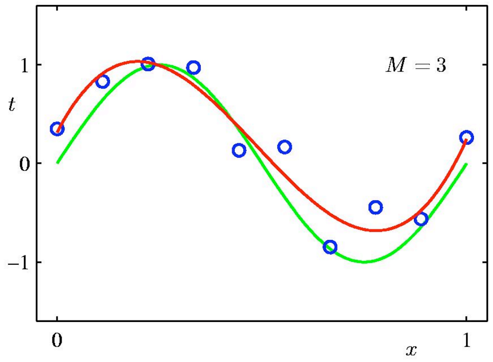
Sample Example (II)

- Polynomial Curve Fitting
  - 1st order polynomial



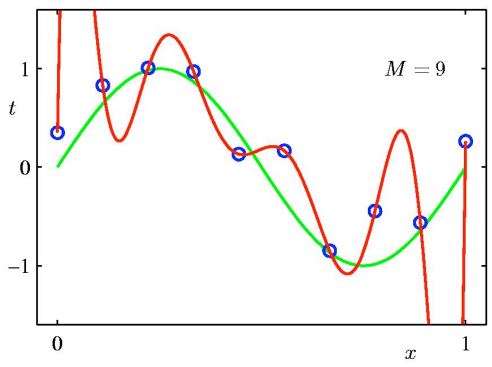
Sample Example (II)

- Polynomial Curve Fitting
  - 3rd order polynomial



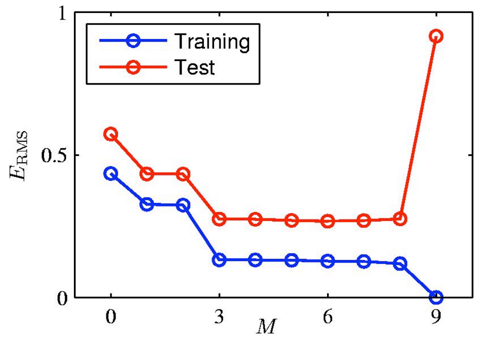
Sample Example (II)

- Polynomial Curve Fitting
  - 9th order polynomial



Sample Example (II)

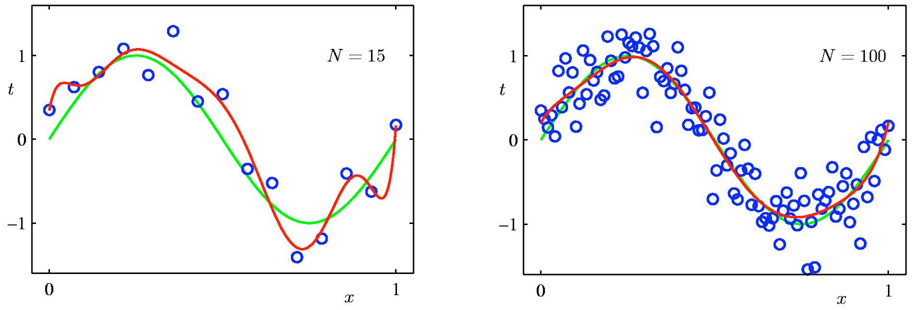
- Polynomial Curve Fitting
  - Over-fitting



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

Sample Example (II)

- Polynomial Curve Fitting
  - Over-fitting and regularization
  - Effect of data set size (9th order polynomial)

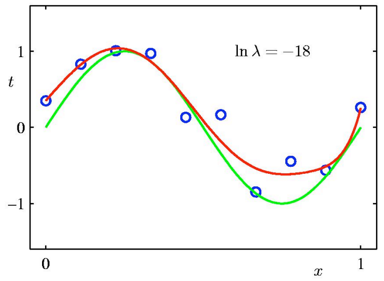


Sample Example (II)

- Polynomial Curve Fitting
  - Regularization
  - Penalize large coefficient values

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

  - 9th order polynomial with $\ln \lambda = -18$



Sample Example (III)

- I want my computer to learn the “style” of Bach and to generate new Bach’s music that has not happened before.
  - Harder to imagine..
  - But we’ll soon get there!

Improvisation Session after learning on Invention No.3 by J.S.Bach

Important Questions

- Given that we have learned what we want...
  - If my *g(x)* can predict well on the data I have, will it also predict well on other sources of *X* I have not seen before?
  - OR To what extent the *knowledge* that has been learned applies to the whole world outside? OR how does my learning *generalize* itself? (**Generalization**)
  - Does having more data necessarily mean I learn better?
    - No! ***Overfitting... .***
  - Does having more complex models necessarily improve learning?
    - No! ***Regularization... .***

Machine Learning Families

Imagine an organism or machine which experiences a series of sensory inputs:

$$x_1, x_2, x_3, x_4, \ldots$$

Supervised learning: The machine is also given desired outputs $y_1, y_2, \ldots$, and its goal is to learn to produce the correct output given a new input.

Unsupervised learning: The goal of the machine is to build a model of x that can be used for reasoning, decision making, predicting things, communicating etc.

Reinforcement learning: The machine can also produce actions $a_1, a_2, \ldots$ which affect the state of the world, and receives rewards (or punishments) $r_1, r_2, \ldots$. Its goal is to learn to act in a way that maximises rewards in the long term.

Machine Learning Families

- Supervised Learning Families:

  Classification: The desired outputs y_i are discrete class labels. The goal is to classify new inputs correctly (i.e. to generalize).

  Regression: The desired outputs y_i are continuous valued. The goal is to predict the output accurately for new inputs.

Machine Learning Models

1. Generative Learning
   - When we start with the hypothesis that a family of parametric models can *generate* X given Y
   - The notion of *prior model!*
   - At the core of Bayesian learning.. Subject of ongoing and historical philosophical debates.
   - Pros:
     - We can incorporate our own belief and knowledge into the model and eventually test and refine it.
     - In most cases simplifies the mathematical structure of the problem.
     - Guaranteed solutions exist in many situations!
   - Cons:
     - Tautology?!
     - *Curse of Dimensionality*

Machine Learning Models

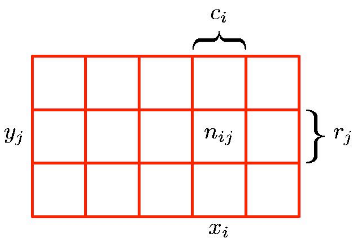
1. Discriminative Learning
   - When we do not assume a model over data, but assume a form on how they are separated from each other and fit it to discriminate classes.... .
   - Neural Networks, Kernel methods, Support Vector Machines etc.
   - Pros:
     - No curse of dimensionality (in most cases)
     - Good when you can not formally describe the hidden generative process.
   - Cons:
     - Prior knowledge for discriminant factors are hard to imagine/justify.. .
     - For complicated problems, they "seem" less intuitive than Generative methods.... .
     - Less appealing for applications where *generation* is also important.... .

Probability Theory

Probability Theory

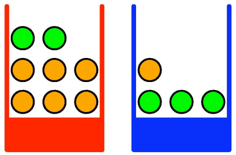
- A probabilistic model of the data can be used to
  - Make inference about missing inputs
  - Generate predictions/fantasies!
  - Make decisions with minimized expected loss
  - Communicate the data in an efficient way
- Statistical modeling is equivalent to other views of learning
  - *Information theoretic:* Finding compact representations of the data
  - *Physics:* Minimizing free energy of a corresponding mechanical system
- If not, what else?
  - *knowledge engineering approach vs. Empirical induction approach*
  - Domain of Probabilities vs. Domain of Possibilities (*fuzzy logic*)
  - Logic AI ...

Probability Theory



Marginal Probability

$$p(X = x_i) = \frac{c_i}{N}.$$



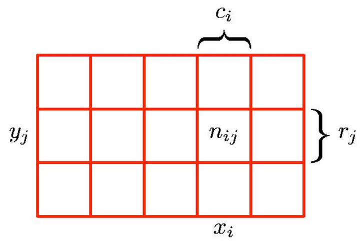
Joint Probability

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

Conditional Probability

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

Probability Theory



Sum Rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N}\sum_{j=1}^{L} n_{ij}$$
$$= \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

Product Rule

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N}$$
$$= p(Y = y_j | X = x_i) p(X = x_i)$$

Probability Theory

- Rules of Probability:

Sum Rule $$p(X) = \sum_Y p(X, Y)$$

Product Rule $$p(X, Y) = p(Y|X)p(X)$$

- Independence:
  - Random variables X and Y are independent if

$$p(X/Y) = p(X)$$

Probability Theory

- Bayes' Theorem:

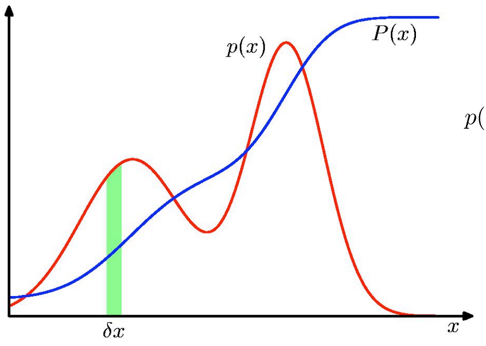
$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

$$p(X) = \sum_Y p(X|Y)p(Y)$$

posterior ∝ likelihood × prior

Probability Theory

- Probability Densities



$$p(x \in (a,b)) = \int_a^b p(x)\,\mathrm{d}x$$

$$P(z) = \int_{-\infty}^{z} p(x)\,\mathrm{d}x$$

$$p(x) \geqslant 0 \qquad \int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1$$

Probability Theory

- Expectations

$$\mathbb{E}[f] = \sum_x p(x) f(x) \qquad \mathbb{E}[f] = \int p(x) f(x)\,\mathrm{d}x$$

$$\mathbb{E}_x[f|y] = \sum_x p(x|y) f(x)$$

Conditional Expectation
(discrete)

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

Approximate Expectation
(discrete and continuous)

Probability Theory

- Variances and Covariances

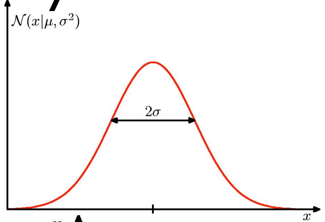
$$\mathrm{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

$$\begin{aligned}\mathrm{cov}[x,y] &= \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y]\end{aligned}$$

$$\begin{aligned}\mathrm{cov}[\mathbf{x},\mathbf{y}] &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^{\mathrm{T}} - \mathbb{E}[\mathbf{y}^{\mathrm{T}}]\}\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^{\mathrm{T}}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^{\mathrm{T}}]\end{aligned}$$

Probability Theory

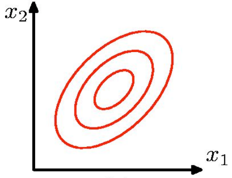
- The Gaussian Distribution



$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$



$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

- Gaussian Mean and Variance:

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) x\,\mathrm{d}x = \mu$$

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) x^2\,\mathrm{d}x = \mu^2 + \sigma^2$$

$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

Basic Rules of Probability

Probabilities are non-negative $P(x) \geq 0 \; \forall x$.

Probabilities normalise: $\sum_x P(x) = 1$ for discrete distributions and $\int p(x)dx = 1$ for probability densities.

The joint probability of x and y is: $P(x, y)$.

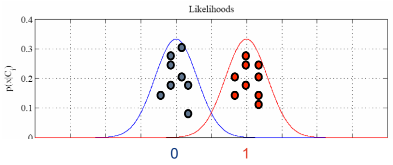
The marginal probability of x is: $P(x) = \sum_y P(x, y)$.

The conditional probability of x given y is: $P(x|y) = P(x, y)/P(y)$
Bayes Rule:

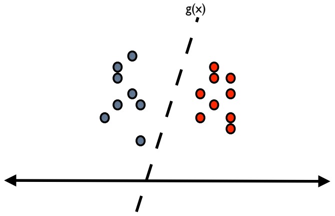
$$P(x, y) = P(x)P(y|x) = P(y)P(x|y) \quad \Rightarrow \quad \boxed{P(y|x) = \frac{P(x|y)P(y)}{P(x)}}$$

Generative vs. Discriminative

- Generative approach:
  - Model $\quad p(t, \mathbf{x}) = p(\mathbf{x}|t)p(t)$
  - Use Bayes' theorem $\quad p(t|\mathbf{x}) = \dfrac{p(\mathbf{x}|t)p(t)}{p(\mathbf{x})}$



- Discriminative approach:
  - Model $p(t|\mathbf{x})$ directly



Bayesian Decision Theory

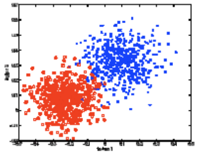
Bayesian Decision Theory

- Framework for computing **optimal decisions** on problems involving **uncertainty** (probabilities)
- Basic concepts:
  - *World:*
    - has states or classes, drawn from a random variable Y
    - Instrument classification, $\quad Y \in \{violin, piano, trumpet, drums, ...\}$
    - Audio to Score Alignment, $\quad Y \in \{note1, chord2, note3, trill4, ...\}$
  - *Observer:*
    - Measures *observations (features)*, drawn from a random process X
    - Instrument classification, $\quad X = MFCCfeatures \in \mathbb{R}^n$

Bayesian Decision Theory

- we will focus on classification problems
  - the observer tries to infer the state of the world

$$g(x) = i, \quad i \in \{1, \ldots, M\}$$



  - we will also mostly consider the “0-1” loss function

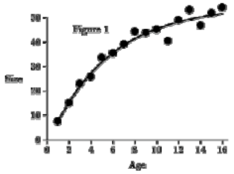
$$\boxed{L[g(x), y] = \begin{cases} 1, & g(x) \neq y \\ 0, & g(x) = y \end{cases}}$$

- but the regression case
  - the observer tries to predict a continuous y

$$g(x) \in \Re$$

  - is basically the same, for a suitable loss function, e.g. squared error

$$\boxed{L[g(x), y] = \|y - g(x)\|^2}$$



Basics of Bayesian Decision Theory

- Question: How to choose the best class given the data?
  - Choose the Maximum A Posteriori (MAP) class:

$$\hat{\omega} = \underset{\omega_i}{\operatorname{argmax}} \, Pr(\omega_i | x)$$

  - Intuitively: Choose the most probable class given the observation.
  - But we don’t know $Pr(\omega_i|x)$
  - But we know $Pr(x|\omega_i)$
  - Apply Bayes rule:

the search for $\hat{\omega} = \underset{\omega_i}{\operatorname{argmax}} \, Pr(\omega_i | x)$

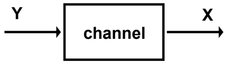
becomes $\underset{\omega_i}{\operatorname{argmax}} \dfrac{p(x|\omega_i) \cdot Pr(\omega_i)}{\sum_j p(x|\omega_j) \cdot Pr(\omega_j)}$

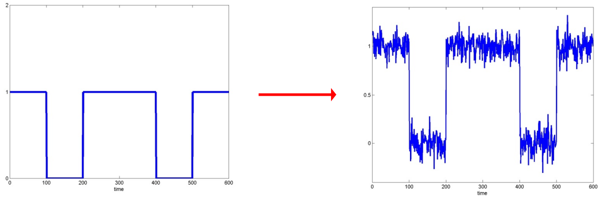
but denominator = $p(x)$ **is the same over all** ω_i

hence $\hat{\omega} = \underset{\omega_i}{\operatorname{argmax}} \, p(x|\omega_i) \cdot Pr(\omega_i)$

Basics of Bayesian Decision Theory

- Let’s now go back to our sample example (I):





- Intuitively, the decision rule can be:

$$Y = \begin{cases} 0, & \text{if } x > T \\ 1, & \text{if } x > T \end{cases}$$

Basics of Bayesian Decision Theory

- We need:
  - Class probabilities:
    - in the absence of any other information let’s say $P_Y(0) = P_Y(1) = 1/2$
  - Class-conditional densities:
    - Noise results from thermal processes, a lot of independent events that add up
    - By the central limit theorem, it is reasonable to assume that noise is Gaussian
- Denote a Gaussian random variable of mean μ and variance by σ

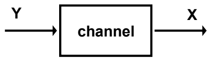
$$X \sim N(\mu, \sigma)$$

Basics of Bayesian Decision Theory

- the Gaussian probability density function is

$$\boxed{P_X(x) = G(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}}$$

- since noise is Gaussian, and assuming it is just added to the signal we have



$$\boxed{X = Y + \varepsilon, \qquad \varepsilon \sim N(0,\sigma^2)}$$

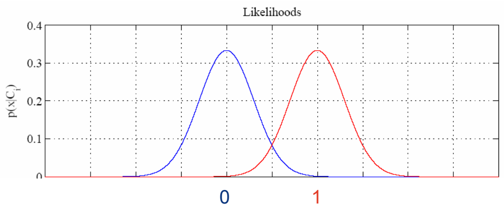
- in both cases, X corresponds to a constant (Y) plus zero-mean Gaussian noise
- this simply adds Y to the mean of the Gaussian

Basics of Bayesian Decision Theory

- In summary:

$$\boxed{\begin{aligned} P_{X|Y}(x\,|\,0) &= G(x,0,\sigma) \\ P_{X|Y}(x\,|\,1) &= G(x,1,\sigma) \end{aligned}} \qquad \boxed{P_Y(0) = P_Y(1) = \tfrac{1}{2}}$$

- or, graphically,



Basics of Bayesian Decision Theory

- To compute the Bayesian Decision Rule (or MAP) we use *log* probabilities here:

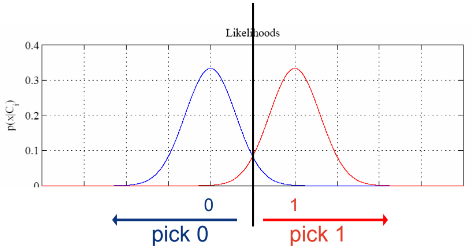
$$\boxed{i^*(x) = \arg\max_i \left[\log P_{X|Y}(x|i) + \log P_Y(i)\right]}$$

- and note that
  - terms which are constant can be dropped
  - Hence, if priors are equal, then we have:

$$i^*(x) = \arg\max_i \log P_{X|Y}(x|i)$$

Basics of Bayesian Decision Theory

- Graphically this MAP solution is equal to:
  - we pick the class that "best explains" (gives higher probability) the observation
  - in this case, we can solve visually



  - but the mathematical solution is equally simple

Basics of Bayesian Decision Theory

- Now let's consider the general case:

$$\boxed{P_{X|Y}(x\,|\,0)=G(x,\mu_0,\sigma) \qquad P_{X|Y}(x\,|\,1)=G(x,\mu_1,\sigma)}$$

  - for which

$$\begin{aligned} i^*(x) &= \arg\max_i \log P_{X|Y}(x\,|\,i) \\ &= \arg\max_i \log\left\{\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu_i)^2}{2\sigma^2}}\right\} \\ &= \arg\max_i \left\{-\frac{1}{2}\log(2\pi\sigma^2)-\frac{(x-\mu_i)^2}{2\sigma^2}\right\} \\ &= \arg\min_i \frac{(x-\mu_i)^2}{2\sigma^2} \end{aligned}$$

Basics of Bayesian Decision Theory

- or

$$\begin{aligned} i^* &= \arg\min_i \frac{(x-\mu_i)^2}{2\sigma^2} \\ &= \arg\min_i (x^2-2x\mu_i+\mu_i^2) \\ &= \arg\min_i (-2x\mu_i+\mu_i^2) \end{aligned}$$

- the optimal decision is, therefore
  - pick 0 if

$$-2x\mu_0+\mu_0^2 < -2x\mu_1+\mu_1^2$$

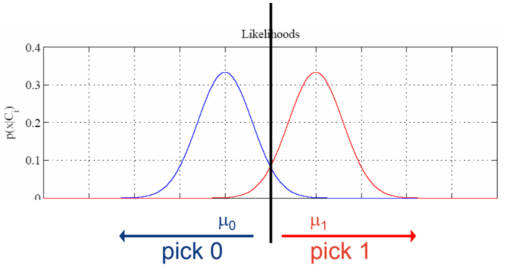
$$2x(\mu_1-\mu_0) < \mu_1^2-\mu_0^2$$

  - or, pick 0 if

$$\boxed{x<\frac{\mu_1+\mu_0}{2}}$$

Basics of Bayesian Decision Theory

- Or graphically,



Basics of Bayesian Decision Theory

- *"Yeah! So what?"*
- Bayesian Decision Theory keeps you *honest!*
  - In practice we never have one variable but a vector of observations >> Multivariate Gaussians
  - Priors are not uniform.
- It also forces us to make our assumptions explicit!
  - assumptions we have made
    - uniform class probabilities $\boxed{P_Y(0)=P_Y(1)=\frac{1}{2}}$
    - Gaussianity $\boxed{P_{X|Y}(x\,|\,i)=G(x,\mu_i,\sigma_i)}$
    - the variance is the same under the two states $\boxed{\sigma_i=\sigma,\forall i}$
    - noise is additive $\boxed{X=Y+\varepsilon}$
  - even for a trivial problem, we have made lots of assumptions

Gaussian Classifiers

- Let's imagine the Bayesian Decision Rule (BDR) for the case of two multivariate Gaussian case:

$$\boxed{P_{X|Y}(x\,|\,i) = \frac{1}{\sqrt{(2\pi)^d\,|\Sigma_i|}}\exp\left\{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right\}}$$

  - the BDR

$$\boxed{i^*(x) = \arg\max_i\left[\log P_{X|Y}(x\,|\,i) + \log P_Y(i)\right]}$$

  - becomes

$$\boxed{\begin{aligned} i^*(x) = \arg\max_i\Big[&-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\\ &-\frac{1}{2}\log(2\pi)^d|\Sigma_i| + \log P_Y(i)\Big]\end{aligned}}$$

Gaussian Classifiers

- this can be written as

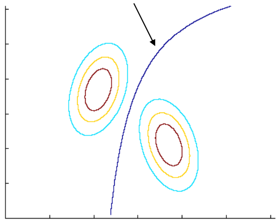
$$\boxed{i^*(x) = \arg\min_i\left[d_i(x,\mu_i) + \alpha_i\right]}$$

with

$$\boxed{d_i(x,y) = (x-y)^T\Sigma_i^{-1}(x-y)}$$

$$\boxed{\alpha_i = \log(2\pi)^d|\Sigma_i| - 2\log P_Y(i)}$$

discriminant: $P_{Y|X}(1|\mathbf{x}) = 0.5$

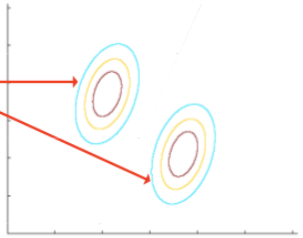


- the optimal rule is to assign x to the closest class
- closest is measured with the Mahalanobis distance $d_i(x,y)$
- to which the α constant is added to account for the class prior

Gaussian Classifiers

- first special case of interest:
  - all classes have the same covariance,

$$\boxed{\Sigma_i = \Sigma, \quad \forall i}$$



- the BDR becomes

$$\boxed{i^*(x) = \arg\min_i\left[d(x,\mu_i) + \alpha_i\right]}$$

  - with

$$\boxed{d(x,y) = (x-y)^T\Sigma^{-1}(x-y)}$$

same metric for all classes

$$\boxed{\alpha_i = \log(\cancel{2\pi})^d|\Sigma| - 2\log P_Y(i)}$$

constant, not function of i, can be dropped

Gaussian Classifiers

- In detail:

$$\begin{aligned} i^*(x) &= \arg\min_i\left[(x-\mu_i)^T\Sigma^{-1}(x-\mu_i) - 2\log P_Y(i)\right]\\ &= \arg\min_i\left[x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_i - \mu_i^T\Sigma^{-1}x + \mu_i^T\Sigma^{-1}\mu_i - 2\log P_Y(i)\right]\\ &= \arg\min_i\left[x^T\Sigma^{-1}x - 2\mu_i^T\Sigma^{-1}x + \mu_i^T\Sigma^{-1}\mu_i - 2\log P_Y(i)\right]\\ &= \arg\max_i\left[\underbrace{\mu_i^T\Sigma^{-1}}_{w_i^T}x \underbrace{- \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + 2\log P_Y(i)}_{w_{i0}}\right]\end{aligned}$$

Gaussian Classifiers

- in summary,

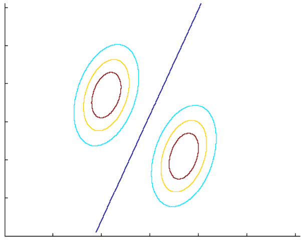
$$\boxed{i^*(x) = \arg\max_i g_i(x)}$$

- with

$$\boxed{\begin{aligned} g_i(x) &= w_i^T x + w_{i0} \\ w_i &= \Sigma^{-1}\mu_i \\ w_{i0} &= -\frac{1}{2}\mu_i^T \Sigma^{-1}\mu_i + \log P_Y(i) \end{aligned}}$$

- the BDR is a linear function or a linear discriminant

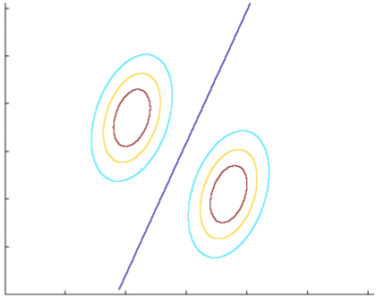
discriminant:
$P_{Y|X}(1|\mathbf{x}) = 0.5$



Gaussian Classifiers

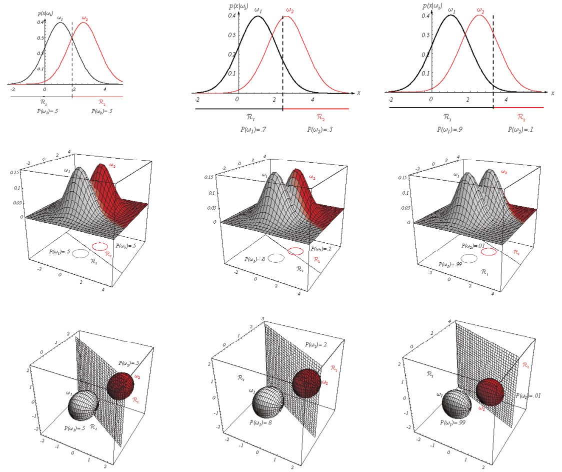
- Group Homework 0
  - Find the Geometric equation for the hyperplane separating the two classes for the linear discriminant of the Gaussian classifier.
  - Hint: This is the set such that

$$\boxed{g_i(x) = g_j(x)}$$



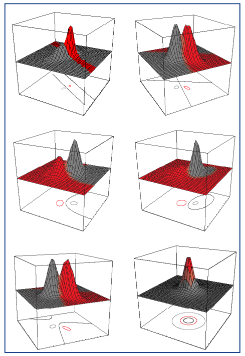
The role of prior

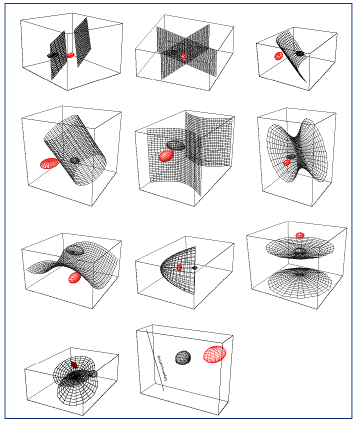
- The prior can offset the “threshold” value (in our simple example):



The role of covariance

- So far, our covariance matrices were simple! If they are different, the the nice *hyper-plane* for a 2-class problem becomes *hyper-quadratic:*
  - in 2 and 3D:





Bayesian Decision Theory

- Advantages
  - BDR is optimat and can not be beaten!
  - Bayes keeps you honest
  - Models reflect *causal interpretation of the problem*, or how we think!
  - Natural decomposition into "what we knew already" (prior) and "what data tells us" (obs)
  - No need for *heuristics* to combine these two sources of information
  - BDR is intuitive
- Problems
  - BDR is optimal ONLY if the *models are correct!*

Maximum Likelihood Estimation

Bayesian Decision Theory

- Advantages
  - BDR is optimal and can not be beaten!
  - Bayes keeps you honest
  - Models reflect causal interpretation of the problem, or how we think!
  - Natural decomposition into "what we knew already" (prior) and "what data tells us" (obs)
  - No need for heuristics to combine these two sources of information
  - BDR is intuitive
- Problems
  - BDR is optimal ONLY if the models are correct!

Bayesian Decision Theory

- WHAT???
  - We do have an optimal (and geometric) solution:

$$i^*(x) \quad = \quad \arg\max_i [\underbrace{\mu_i^T \Sigma^{-1} x}_{w_i^T} - \underbrace{\frac{1}{2}\mu_i^T \Sigma^{-1}\mu_i + 2\log P_Y(i)]}_{w_{i0}}$$

  - but we do not know the values of the parameters $\mu, \Sigma, P_Y(i)$
  - We have to *estimate* these values!
  - We can estimate from a training set
    - example: use the average value as an estimate for the mean!

Maximum Likelihood

- We rely on the maximum likelihood (ML) principle.
- ML has three main steps:
  1. Choose a parametric model for all probabilities (as a function of unknown parameters)
  2. Assemble a training data-set
  3. Solve for parameters that maximize probabilities on the data-set!

Maximum Likelihood

- we rely on the maximum likelihood (ML) principle
- this has three steps:
  - 1) we choose a parametric model for all probabilities
  - to make this clear we denote the vector of parameters by Θ and the class-conditional distributions by

$$\boxed{P_{X|Y}(x|i;\Theta)}$$

  - note that this means that Θ is NOT a random variable (otherwise it would have to show up as subscript)
  - it is simply a parameter, and the probabilities are a function of this parameter

Maximum Likelihood

- 2) we assemble a collection of datasets $\mathcal{D}^{(i)} = \{x_1^{(i)}, \ldots, x_n^{(i)}\}$ set of examples drawn independently from class i
- 3) we select the parameters of class i to be the ones that maximize the probability of the data from that class

$$\boxed{\begin{aligned}\Theta_i &= \arg\max_{\Theta} P_{X|Y}\left(D^{(i)}\,|\,i;\Theta\right)\\ &= \arg\max_{\Theta} \log P_{X|Y}\left(D^{(i)}\,|\,i;\Theta\right)\end{aligned}}$$

- like before, it does not really make any difference to maximize probabilities or their logs

Maximum Likelihood

- Maximum Likelihood
  - since
    - each sample $\mathcal{D}^{(i)}$ is considered independently
    - parameter Θ_i estimated only from sample $\mathcal{D}^{(i)}$
  - we simply have to repeat the procedure for all classes
  - so, from now on we omit the class variable

$$\boxed{\begin{aligned}\Theta^* &= \arg\max_{\Theta} P_X\left(D;\Theta\right)\\ &= \arg\max_{\Theta} \log P_X\left(D;\Theta\right)\end{aligned}}$$

  - the function $P_X(\mathcal{D};\Theta)$ is called the likelihood of the parameter Θ with respect to the data
  - or simply the likelihood function

Maximum Likelihood

- *In short:* Given some data-points, we are solving for

$$\boxed{\Theta^* = \underset{\Theta}{\arg\max}\, P_X\left(D;\Theta\right)}$$

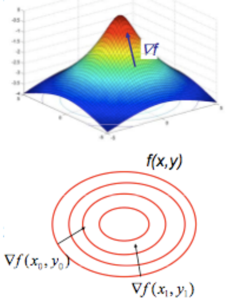
- If Θ is scalar, this is high-school calculus!
  - We have maximum when first derivative is zero + second derivative is negative!
- We review higher dimensional tools for this aim very quick... .

Maximum Likelihood

- The gradient:
  - in higher dimensions, the generalization of the derivative is the gradient. The gradient of a function f(x) at z is:

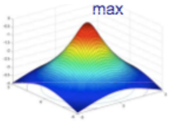
$$\nabla f(z) = \left(\frac{\partial f}{\partial x_0}(z), \ldots, \frac{\partial f}{\partial x_{n-1}}(z)\right)^T$$

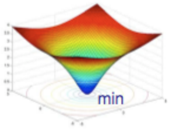
  - It has a nice geometric interpretation:
    - It points in the direction of *maximum growth* of the function
    - *Perpendicular* to the contour where the function is constant

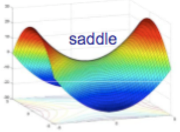


Maximum Likelihood

- The gradient
  - note that if $\nabla f = 0$
    - there is no direction of growth
    - also $-\nabla f = 0$, and there is no direction of decrease
    - we are either at a local minimum or maximum or “saddle” point
  - conversely, at local min or max or saddle point
    - no direction of growth or decrease
    - $\nabla f = 0$
  - this shows that we have a critical point if and only if $\nabla f = 0$
  - to determine which type we need second order conditions







Maximum Likelihood

- The Hessian:
  - extension of the 2nd-order derivative is the Hessian Matrix:

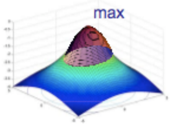
$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_0^2} & \frac{\partial^2 f}{\partial x_0\, \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_0\, \partial x_{n-1}} \\ \frac{\partial^2 f}{\partial x_1\, \partial x_0} & \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1\, \partial x_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_{n-1}\, \partial x_0} & \frac{\partial^2 f}{\partial x_{n-1}\, \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_{n-1}^2} \end{bmatrix}$$

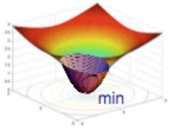
  - In an ML setup we have a *maximum* when Hessian is negative definite or

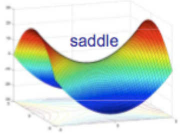
$$x^T \nabla^2 f(x) x \leq 0$$

Maximum Likelihood

- The Hessian:
  - this means that, when gradient is zero at x, we have
    - a maximum when function can be approximated by an "upwards-facing" quadratic
    - a minimum when function can be approximated by a "downwards-facing" quadratic
    - a saddle point otherwise







Maximum Likelihood

- In summary:
  1. Choose a parametric model for probabilities $P_X(x;\Theta)$
  2. Assemble $D = \{X_1, \ldots, X_n\}$ of independently drawn examples
  3. Select parameters that maximize the probability of the data
     - or Given a data-set we need to solve

$$\boxed{\begin{aligned}\Theta^* &= \arg\max_{\Theta} P_X(D;\Theta) \\ &= \arg\max_{\Theta} \log P_X(D;\Theta)\end{aligned}}$$

     - The solutions are the parameters such that

$$\begin{aligned}\nabla_\Theta P_X(D;\Theta) &= 0 \\ \theta^t \nabla^2_\Theta P_X(D;\theta)\theta &\leq 0, \forall\theta \in \mathbb{R}^n\end{aligned}$$

Maximum Likelihood

- Sample Example II: *Polynomial Regression*
  - Two random variables X and Y
  - A dataset of examples

$$D = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

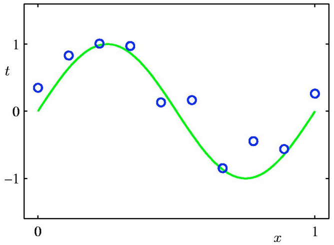
  - A parametric model of the form

$$y = f(x;\Theta) + \epsilon \text{ where } \epsilon \sim N(0,\sigma^2)$$

  - Concretely, $f(x;\Theta) = \sum_{i=0}^{K} \theta_i x^i$
  - where the data is distributed as $P_{Z|X}(D|x;\Theta) = G(z, f(x;\Theta), \sigma^2)$
  - Show that $\Theta^* = [\Gamma^T\Gamma]^{-1}\Gamma^T y$ where

$$\Gamma = \begin{bmatrix} 1 & \cdots & x_1^K \\ \vdots & \vdots & \vdots \\ 1 & \cdots & x_n^K \end{bmatrix}$$

- GROUP I (for next class)

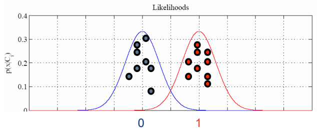


ML + BDR

- Going back to our simple classification problem... .
  - We can combine ML and Bayesian Decision Rule to make things safer and pick the desired class *i* if:

$$i^*(x) = \arg\max_i P_{X|Y}(x|i;\theta_i^*)P_Y(i)$$

$$\text{where } \theta_i^* = \arg\max_\theta P_{X|Y}(D|i,\theta)$$



Estimators

- We now know how to produce estimators using Maximum-Likelihood... .
- How do we evaluate an estimator? Using bias and variance
- Bias
  - A measure how the expected value is equal to the true value
  - If $\hat{\theta} = f(X_1, \ldots, X_n)$ then $Bias(\hat{\theta}) = E_{X_1,\ldots,X_n}[f(X_1, \ldots, X_n) - \theta]$
  - An estimator that has bias will usually not converge to the perfect estimate! No matter how large the data-set is!
- Variance
  - Given a good bias, how many sample points do we need?
  - $Var(\hat{\theta}) = E_{X_1,\ldots,X_n}\left\{f(X_1, \ldots, X_n) - E_{X_1,\ldots,X_n}[f(X_1, \ldots, X_n)]^2\right\}$
  - Variance usually decreases with more training examples... .

Estimators

- Example
  - ML estimator for the mean of a Gaussian $N(\mu, \sigma^2)$

$$\begin{aligned} Bias(\hat{\mu}) &= E_{X_1,\ldots,X_n}[\hat{\mu} - \mu] \\ &= E_{X_1,\ldots,X_n}[\hat{\mu}] - \mu \\ &= \frac{1}{n}\sum_i E_{X_1,\ldots,X_n}[X_i] - \mu \\ &= \frac{1}{n}\sum_i E_{X_i}[X_i] - \mu \\ &= \mu - \mu = 0 \end{aligned}$$

  - The estimator is thus *unbiased*

Bayesian Parameter Estimation

Bayesian Parameter Estimation

- Bayesian parameter estimation is an alternative framework for parameter estimation
- There is fundamental difference between Bayesian and ML methods!
  - The long debate between *frequentists vs Bayesians*
  - To understand this, we need to distinguish between two components:
    - The definition of probability (intact)
    - The assessment of probability (differs)
  - We need to review these fundamentals!

Probability Measures

- This does not change between *frequentist* and *Bayesian* philosophies
- Probability measure satisfies three axioms:
  - $P(A) \geq 0 \quad \forall \text{events A}$
  - $P(\text{universal event}) = 1$
  - $\text{if } A \bigcap B = \emptyset \;\text{ then } P(A+B) = P(A) + P(B)$

Frequentist vs Bayesian

- Difference is in interpretation!
- Frequentist view:
  - Probabilities are relative frequencies
  - Make sense when we have a lot of observations (no bias)
  - Problems:
    - In most cases we do not have large number of observations!
    - In most cases probabilities are not *objective*!
    - This is not usually how people behave.
- Bayesian view:
  - Probabilities are *subjective* (not equal to relative count)
  - Probabilities are *degrees of belief* on the outcome of experiment

Bayesian Parameter Estimation

- Difference with ML: Θ is a random variable.
- Basic concept:
  - Training set $D = \{X_1, \ldots, X_n\}$ of examples drawn independently
  - Probability density for observations given parameter

    $$\boxed{P_{X|\Theta}(x|\Theta)}$$

  - Prior distribution for parameter configurations

    $$\boxed{P_\Theta(\theta)}$$

    encodes prior belief on Θ
  - **Goal:** Compute the posterior distribution

    $$\boxed{P_{\Theta|X}(\Theta|D)}$$

Bayes vs. ML

- Optimal estimate
  - under ML there is one "best" estimate
  - under Bayes there is no "best" estimate
  - It makes no sense under Bayes to talk about "best" estimate
- Predictions
  - We do not really care about the parameters themselves! Only in the fact that they build models... .
  - Models can be used to make predictions
  - Unlike ML, Bayes uses *ALL* information in the training set to make predictions

Bayes vs ML

- let's consider the BDR under the "0-1" loss and an independent sample $\mathcal{D} = \{x_1 , ..., x_n\}$
- ML-BDR:
  - pick i if

$$\boxed{\begin{aligned} i^*(x) &= \arg\max_i P_{X|Y}\left(x \mid i;\theta_i^*\right) P_Y(i) \\ \text{where } \theta_i^* &= \arg\max_\theta P_{X|Y}\left(D \mid i,\theta\right) \end{aligned}}$$

- two steps:
  - i) find θ^*
  - ii) plug into the BDR
- all information not captured by θ^* is lost, not used at decision time

Bayes vs ML

- note that we know that information is lost
  - e.g. we can't even know how good of an estimate θ^* is
  - unless we run multiple experiments and measure bias/variance
- Bayesian BDR
  - under the Bayesian framework, everything is conditioned on the training data
  - denote $T = \{X_1 , ..., X_n\}$ the set of random variables from which the training sample $\mathcal{D} = \{x_1 , ..., x_n\}$ is drawn
- B-BDR:
  - pick i if

$$\boxed{i^*(x) = \arg\max_i P_{X|Y,T}\left(x \mid i, D_i\right) P_Y(i)}$$

- the decision is conditioned on the entire training set

Bayesian BDR

- to compute the conditional probabilities, we use the marginalization equation

$$P_{X|Y,T}\left(x \mid i, D_i\right) = \int P_{X|\Theta,Y,T}\left(x \mid \theta, i, D_i\right) P_{\Theta|Y,T}\left(\theta \mid i, D_i\right) d\theta$$

- note 1: when the parameter value is known, x no longer depends on T, e.g. $X|\Theta \sim N(\theta,\sigma^2)$
  - we can, simplify equation above into

$$\boxed{P_{X|Y,T}\left(x \mid i, D_i\right) = \int P_{X|\Theta,Y}\left(x \mid \theta, i\right) P_{\Theta|Y,T}\left(\theta \mid i, D_i\right) d\theta}$$

- note 2: once again can be done in two steps (per class)
  - i) find $P_{\Theta|T}(\theta|D_i)$
  - ii) compute $P_{X|Y,T}(x|i, D_i)$ and plug into the BDR
- no training information is lost

Bayesian BDR

- in summary
  - pick i if

$$\boxed{\begin{aligned} & i^*(x) = \arg\max_i P_{X|Y,T}\left(x \mid i, D_i\right) P_Y(i) \\ & \textit{where} \quad P_{X|Y,T}\left(x \mid i, D_i\right) = \int P_{X|Y,\Theta}\left(x \mid i, \theta\right) P_{\Theta|Y,T}\left(\theta \mid i, D_i\right) d\theta \end{aligned}}$$

- note:
  - as before the bottom equation is repeated for each class
  - hence, we can drop the dependence on the class
  - and consider the more general problem of estimating

$$\boxed{P_{X|T}\left(x \mid D\right) = \int P_{X|\Theta}\left(x \mid \theta\right) P_{\Theta|T}\left(\theta \mid D\right) d\theta}$$

Predictive Distribution

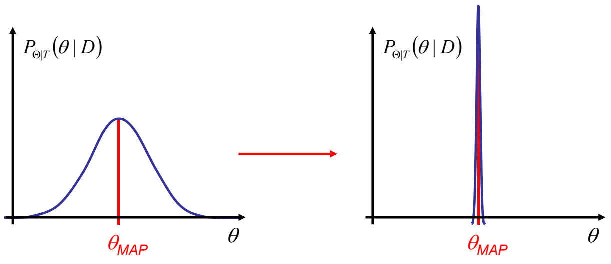
- The distribution
  $$P_{X|T}(x|D) = \int P_{X|\Theta}(x|\theta)P_{\Theta|T}(\theta|D)d\theta$$
  is known as the predictive distribution. It allows us
  - to predict the value of x given ALL the information in the training set
- Bayes vs. ML:
  - ML picks *one* model, Bayes averages all models
  - ML is a special case of Bayes when we are *very confident* about the model
  - In otherwords ML~Bayes when
    - prior is narrow
    - if the sample space is quite large
    - intuition: Given a lot of training data, there is little uncertainty
  - Bayes regularizes the ML estimate!

MAP approximation

- this sounds good, why use ML at all?
- the main problem with Bayes is that the integral
  $$\boxed{P_{X|T}(x|D) = \int P_{X|\Theta}(x|\theta)P_{\Theta|T}(\theta|D)d\theta}$$
  can be quite nasty
- in practice one is frequently forced to use approximations
- one possibility is to do something similar to ML, i.e. pick only one model
- this can be made to account for the prior by
  - picking the model that has the largest posterior probability given the training data
  $$\boxed{\theta_{MAP} = \arg\max_{\theta} P_{\Theta|T}(\theta|D)}$$

MAP approximation

- this can usually be computed since
  $$\boxed{\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} P_{\Theta|T}(\theta|D)\\ &= \arg\max_{\theta} P_{T|\Theta}(D|\theta)P_{\Theta}(\theta)\end{aligned}}$$
  and corresponds to approximating the prior by a delta function centered at its maximum



MAP approximation

- in this case
  $$\boxed{\begin{aligned}P_{X|T}(x|D) &= \int P_{X|\Theta}(x|\theta)\delta(\theta-\theta_{MAP})d\theta\\ &= P_{X|\Theta}(x|\theta_{MAP})\end{aligned}}$$
- the BDR becomes
  - pick i if
  $$\boxed{\begin{aligned}&i^*(x) = \arg\max_{i} P_{X|Y}(x|i;\theta_i^{MAP})P_Y(i)\\ &\text{where } \theta_i^{MAP} = \arg\max_{\theta} P_{T|Y,\Theta}(D|i,\theta)P_{\Theta|Y}(\theta|i)\end{aligned}}$$
  - when compared to the ML this has the advantage of still accounting for the prior (although only approximately)

MAP vs ML

- ML-BDR
  - pick i if

$$\boxed{\begin{aligned} & i^*(x) = \arg\max_i P_{X|Y}\left(x \mid i; \theta_i^*\right) P_Y(i) \\ & \text{where } \theta_i^* = \arg\max_\theta P_{X|Y}\left(D \mid i, \theta\right) \end{aligned}}$$

- Bayes MAP-BDR
  - pick i if

$$\boxed{\begin{aligned} & i^*(x) = \arg\max_i P_{X|Y}\left(x \mid i; \theta_i^{MAP}\right) P_Y(i) \\ & \text{where } \theta_i^{MAP} = \arg\max_\theta P_{T|Y,\Theta}\left(D \mid i, \theta\right) P_{\Theta|Y}\left(\theta \mid i\right) \end{aligned}}$$

  - the difference is non-negligible only when the dataset is small
- there are better alternative approximations

Bayesian Learning

Summary

Apply the basic rules of probability to learning from data.
Data set: $\mathcal{D} = \{x_1, \ldots, x_n\}$ Models: m, m' etc. Model parameters: θ

Prior probabilities on models: $P(m)$, $P(m')$ etc.

Prior probabilities on model parameters: e.g. $P(\theta|m)$

Model of data given parameters: $P(x|\theta, m)$

If the data are independently and identically distributed then:

$$P(\mathcal{D}|\theta, m) = \prod_{i=1}^{n} P(x_i|\theta, m)$$

Posterior probability of model parameters:

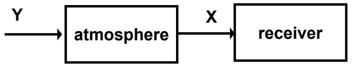
$$P(\theta|\mathcal{D}, m) = \frac{P(\mathcal{D}|\theta, m)P(\theta|m)}{P(\mathcal{D}|m)}$$

Posterior probability of models:

$$P(m|\mathcal{D}) = \frac{P(m)P(\mathcal{D}|m)}{P(\mathcal{D})}$$

Example

- communications problem



Y atmosphere X receiver

- two states:
  - Y=0 transmit signal s = $-\mu_0$
  - Y=1 transmit signal s = μ_0
- noise model

$$\boxed{X = Y + \varepsilon, \qquad \varepsilon \sim N(0, \sigma^2)}$$

Example

- the BDR is
  - pick “0” if

$$\boxed{x < \frac{\mu_0 + \left(-\mu_0\right)}{2} = 0}$$

- this is optimal and everything works wonderfully, but
  - one day we get a phone call: the receiver is generating a lot of errors!
  - there is a calibration mode:
    - rover can send a test sequence
    - but it is expensive, can only send a few bits
  - if everything is normal, received means should be μ_0 and $-\mu_0$

Example

- action:
  - ask the system to transmit a few 1s and measure X
  - compute the ML estimate of the mean of X

$$\boxed{\mu = \frac{1}{n}\sum_i X_i}$$

- result: the estimate is different than μ_0
- we need to combine two forms of information
  - our prior is that $\boxed{\mu \sim N(\mu_0,\sigma^2)}$
  - our “data driven” estimate is that

$$\boxed{X \sim N(\hat{\mu},\sigma^2)}$$

Bayesian solution

- Gaussian likelihood (observations)

$$\boxed{P_{T|\mu}(D|\mu) = G(D,\mu,\sigma^2) \qquad \sigma^2 \text{is known}}$$

- Gaussian prior (what we know)

$$\boxed{P_\mu(\mu) = G(\mu,\mu_0,\sigma_0^2)}$$

  - μ_0,σ_0^2 are known hyper-parameters
- we need to compute
  - posterior distribution for μ $\boxed{P_{\mu|T}(\mu|D) = \frac{P_{T|\mu}(D|\mu)P_\mu(\mu)}{P_T(D)}}$

Bayesian solution

- the posterior distribution is

$$\boxed{P_{\mu|T}(\mu|D) = G\left(\mu,\mu_n,\sigma_n^2\right)}$$

$$\mu_n = \frac{\sigma_0^2\sum_i x_i + \mu_0\sigma^2}{\sigma^2+n\sigma_0^2} \Rightarrow \boxed{\mu_n = \underbrace{\frac{n\sigma_0^2}{\sigma^2+n\sigma_0^2}}_{\alpha_n}\mu_{ML} + \underbrace{\frac{\sigma^2}{\sigma^2+n\sigma_0^2}}_{1-\alpha_n}\mu_0}$$

$$\sigma_n^2 = \left(\frac{\sigma^2\sigma_0^2}{\sigma^2+n\sigma_0^2}\right) \Rightarrow \boxed{\frac{1}{\sigma_n^2} = \frac{1}{\sigma_0^2}+\frac{n}{\sigma^2}}$$

- this is intuitive

Bayesian solution

- for free, Bayes also gives us
  - the weighting constants

$$\boxed{\alpha_n = \frac{n\sigma_0^2}{\sigma^2+n\sigma_0^2}}$$

  - a measure of the uncertainty of our estimate

$$\boxed{\frac{1}{\sigma_n^2} = \frac{1}{\sigma_0^2}+\frac{n}{\sigma^2}}$$

  - note that $1/\sigma^2$ is a measure of precision
  - this should be read as

$$P_{Bayes} = P_{ML} + P_{prior}$$

  - Bayesian precision is greater than both that of ML and prior

Observations

- 1) note that precision increases with n, variance goes to zero

$$\boxed{\frac{1}{\sigma_n^2}=\frac{1}{\sigma_0^2}+\frac{n}{\sigma^2}}$$

we are guaranteed that in the limit of infinite data we have convergence to a single estimate

- 2) for large n the likelihood term dominates the prior term

$$\boxed{\begin{aligned}&\mu_n=\alpha_n\hat{\mu}+(1-\alpha_n)\mu_0\\&\qquad\alpha_n\in[0,1],\quad \alpha_n\xrightarrow[n\to\infty]{}1,\quad \alpha_n\xrightarrow[n\to 0]{}0\end{aligned}}$$

the solution is equivalent to that of ML

- for small n, the prior dominates
- this always happens for Bayesian solutions

$$P_{\mu|T}(\mu\,|\,D)\propto\prod_i P_{X|\mu}(x_i\,|\,\mu)P_\mu(\mu)$$

Observations

- 3) for a given n

$$\boxed{\alpha_n=\frac{n\sigma_0^2}{\sigma^2+n\sigma_0^2}}\quad\boxed{\begin{aligned}&\mu_n=\alpha_n\hat{\mu}+(1-\alpha_n)\mu_0\\&\qquad\alpha_n\in[0,1],\quad \alpha_n\xrightarrow[n\to\infty]{}1,\quad \alpha_n\xrightarrow[n\to 0]{}0\end{aligned}}$$

if $\sigma_0^2>>\sigma^2$, i.e. we really don't know what μ is a priori
then $\mu_n = \mu_{ML}$

- on the other hand, if $\sigma_0^2<<\sigma^2$, i.e. we are very certain a priori, then $\mu_n = \mu_0$

- in summary,
  - Bayesian estimate combines the prior beliefs with the evidence provided by the data
  - in a very intuitive manner

Conjugate priors

- note that
  - the prior $P_\mu(\mu)=G(\mu,\mu_0,\sigma_0^2)$ is Gaussian
  - the posterior $P_{\mu|T}(\mu\,|\,D)=G(x,\mu_n,\sigma_n^2)$ is Gaussian
- whenever this is the case (posterior in the same family as prior) we say that
  - $P_\mu(\mu)$ is a conjugate prior for the likelihood $P_{X|\mu}(x\,|\,\mu)$
  - posterior $P_{\mu|T}(\mu\,|\,D)$ is the reproducing density
- a number of likelihoods have conjugate priors

Likelihood	Conjugate prior
Bernoulli	Beta
Poisson	Gamma
Exponential	Gamma
Normal (known σ^2)	Gamma

Group Homework 2

- Histogram Problem
  - Imagine a random variable X such that, $P_X(k)=\pi_k,\ k\in 1,\dots,N$
  - Suppose we draw *n* independent observations from X and form a random vector $C=(C_1,\cdots,C_N)^T$ where C_k is the number of times where the observed value is *k*
  - **C** is then a *histogram* and has a *multinomial distribution:*

$$P_{C_1,\dots,C_N}(c_1,\dots,c_N)=\frac{n!}{\prod_{k=1}^N c_k!}\prod_{j=1}^N\pi_j^{c_j}$$

  - Note that $\pi=(\pi_1,\dots,\pi_w)$ are probabilities and thus: $\pi_i\geq 0$, $\sum\pi_i=1$

1. Derive the ML estimate for parameters $\pi_k,\ k\in\{1,...,N\}$
   - *hint:* If you know about lagrange multipliers, use them! Otherwise, keep in mind that minimizing for a function *f(a,b)* constraint to *a+b=1* is equivalent to minimizing for *f(a,1-a)*.

Group Homework 2

- Histogram Problem

2. Derive the MAP solution using Dirichlet priors:
   - One possible prior model over π_k is the *Dirichlet Distribution:*

$$P_{\Pi_1,\ldots,\Pi_N}(\pi_1,\ldots,\pi_N) = \frac{\Gamma(\sum_{j=1}^{N})u_j)}{\prod_{j=1}^{N}\Gamma(u_j)}\prod_{j=1}^{N}\pi_j^{u_j-1}$$

   - where *u* is the set of *hyper-parameters* (prior parameters to solve) and

$$\Gamma(x) = \int_O^{\infty} e^{-t}t^{x-1}dt$$

   is the Gamma function.

   - You should show that the posterior is equal to:

$$P_{\Pi|C}(\pi|c) = \frac{\Gamma(\sum_{j=1}^{W} c_j + u_j)}{\prod_{k=1}^{W}\Gamma(c_j+u_j)}\prod_{j=1}^{W}\pi_j^{c_j+u_j-1}$$

3. Compare the MAP estimator with that of ML in part (1). What is the role of this prior compared to ML?